(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
2 August 2001 (02.08.2001)

PCT

(10) International Publication Number
WO 01/55813 A2

(51) International Patent Classification[7]: G06F

(21) International Application Number: PCT/US01/00017

(22) International Filing Date: 18 January 2001 (18.01.2001)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
09/484,451 18 January 2000 (18.01.2000) US

(71) Applicant: NOVELL, INC. [US/US]; 1800 South Novell Place, Provo, UT 84606 (US).

(72) Inventor: JNEID, Amer; 31526 Sea Shadows, Laguna Niguel, CA 92677 (US).

(74) Agents: MCCOMBS, David, L. et al.; Haynes and Boone, LLP, 901 Main Street, Suite 3100, Dallas, TX 75202 (US).

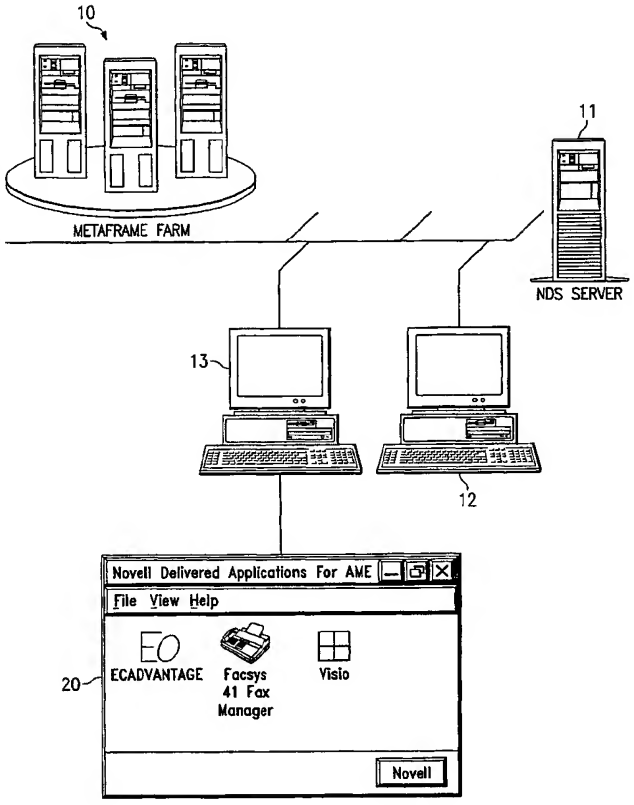
(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CR, CU, CZ, DE, DK, DM, DZ, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, UZ, VN, YU, ZA, ZW.

(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Published:
— *without international search report and to be republished upon receipt of that report*

[Continued on next page]

(54) Title: SYSTEM AND METHOD FOR USING DIRECTORY SERVICES TO FACILITATE ACCESS TO APPLICATIONS AVAILABLE ON THIN CLIENT SERVERS



(57) Abstract: In order to permit a directory service such as Novell Directory Services™ to not only manage application stored locally on a network, but also to manage access to thin client application programs supplied by any number of thin client servers or server farms, the thin client application programs are published in the directory services. During sign-on to the published applications, any information necessary to locate the thin client servers is retrieved from the directory serves and supplied to the client that carries out the sing-on procedure. The thin client application programs may be Citrix Systems' MetaFrame ®* published applications, and the sign-on client may be a Citrix Systems ICA ®* client.

WO 01/55813 A2

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

System And Method For Using Directory Services To Facilitate Access To Applications Available On Thin Client Servers

BACKGROUND OF THE INVENTION

1. Field of the Invention

This invention relates to the field of server-based or "thin client" computing, in which application programs are run on a remote server, and only screen updates, keystrokes, and mouse clicks are transferred between the remote server and the user's computing device.

The system of the invention is especially applicable to application servers which utilize Microsoft's Windows NT Server, Terminal Server Edition, known as Windows Terminal Server, and to Citrix Systems' MetaFrame® application server software.

More specifically, the invention relates

(i) to a system and method for storing and managing information concerning published application programs remotely run on a thin client server (hereinafter referred to as "published applications" or "published application programs") so that a user can retrieve information published application programs from a single source, known as a directory service, and thereby locate and access the published application programs irrespective of the number and actual locations of the thin client servers on which the published application programs are run, and

(ii) to a single sign-on system and method which permits the user to authenticate to the directory service and subsequently access any application listed or "published" on the directory service, whether of the conventional or thin client type.

In the system of the invention, an information package concerning each of the thin client applications to be made available is published in the directory service, and access to the thin client application programs is established by running a first client program associated with the server on which the thin client application is located, and a pre-

launched second client program that supplies sign-on and authentication information from the directory service to the first client program. The first client program may be the Citrix Systems' Independent Computing Architecture (ICA®) client currently used to launch MetaFrame® published applications, or any other published application launching program, while the directory services may be provided by Novell Directory Services™ (NDS®), Novell ZENworks® (which integrates NDS®), or by any other directory services application program capable of carrying out the functions described below.

2. Description of Related Art

As anyone who manages computer networks or systems for an organization is aware, the task of keeping up with the latest software updates, and of keeping hardware current, is becoming increasingly complex. As the organization grows and new computing equipment is added, the number of different types of computing devices inevitably increases because the added equipment will incorporate any changes in hardware or software made since the last addition of equipment. Each of these different computing devices must be supplied with applications programs and program updates tailored to the individual devices, while preferably maintaining overall system compatibility.

One way to reduce costs, ensure software compatibility, and yet provide for the latest updates, is to utilize the concept of server based or thin client computing, in which the end-user's computing device only needs to be able to communicate keyboard entries, mouse clicks, or other input signals to the thin client server, and to update the end-user's display as the application is run, thereby minimizing the cost of the hardware and simplifying software maintenance.

An especially versatile example of such a thin client computing system is Citrix Systems' Independent Computing Architecture (ICA), illustrated in Fig. 1, implemented using Citrix Systems' MetaFrame™ server software, which enables applications to be run remotely through all commonly used communications protocols, including TCP/IP. In principle, the MetaFrame™ server software can supply application programs to any computing device irrespective of configuration including, as shown in Fig. 1, cross-platform (non-Windows™) desktops 2, remote computing devices 3, branch office systems 4, thin client terminals 5, and wireless terminals 6, and through any type of network or combination of networks including, as also shown in Fig. 1, local area networks (LANs) 7, the Internet 8, or corporate wide area networks (WANs) 9.

In practice, however, the ability to supply thin client applications such as MetaFrame® published applications over a conventional system of the type shown in Fig. 1 has been limited by difficulties in broadcasting the availability of the published applications when the conventional system is expanded or scaled to include multiple thin client servers and/or multiple load-balanced groups of servers (known as server farms) at dispersed locations on an open network such as the Internet. Internet servers are currently set up to locate addresses of computers without regard to what is on those computers, while conventional Internet search engines lack the ability to supply the type of information necessary to not only locate server on which a particular published application is available, but to select the server and facilitate sign-on.

Thus, use of thin client technology has been unnecessarily limited, despite its obvious advantages, by the current lack of any way to supply thin client application over the Internet, and in particular by the lack of solutions to the problems of (i) broadcasting the published applications, *i.e.*, informing users how to find and access the thin client servers on which desired published application programs are located, (ii) selecting from among multiple thin client servers on which the

published applications are available, (iii) tracking usage of published applications for billing and/or licensing purposes, and (iv) handling multiple authentication procedures and passwords necessary to access different thin client servers.

Ideally, a user of a thin client application should be able to access published applications in the same manner as an application stored on a local drive, *e.g.*, by clicking on an icon representing the application, without being aware that the application is a thin client application, much less having to locate and select suitable thin client servers, or having to undergo the particular authentication and sign-on procedures required before published applications can actually be accessed on the selected server. Such transparent server location, selection, and sign-on is currently impossible.

In addition to the problems of locating and accessing published applications, many conventional thin client computing arrangements, including MetaFrame® server systems, are subject to security risks resulting from the local storage, in text based configuration files, of sign-on and authentication information such as account numbers and passwords necessary to access the thin client applications. While it is not necessary in these systems to locally store

the sign-on and authentication information, in practice most users elect to store the information in order to simply sign-on by eliminating the need to recall and enter the information each time a published application is to be accessed. In the case of MetaFrame® server software, the text based configuration files are known as appsrv.ini files.

Yet another problem with conventional thin client arrangements is the problem of time zone management, which may arise when the thin client server is in a different time zone than the client computer and the remotely run published application program is a time sensitive program. Although it is currently not practical to run thin client programs from multiple servers or server farms over the Internet, where the time zone problem would be most severe, the problem can nevertheless arise on private corporate networks due to the widely dispersed, and even global nature, of many corporations.

To solve these and other problems, the present invention proposes to use a directory service to "publish" or store information concerning the thin client applications. Directory services are currently used to facilitate management of networks by listing each application and piece of equipment available on the

network, and associating the equipment and applications with users. The equipment and applications are conveniently tracked by a directory service as objects in a directory tree, the directory service essentially maintaining records on each object by tracking changes and updating the object records files as necessary. When a user signs onto the network, the directory service or a related implementation program authenticates the user, and then manages requests for access to the various objects on the network. New software and updates are distributed as necessary based on stored profiles, and the directory service is updated to reflect the changes.

Such directory services are well-known, and provide a convenient centralized way for the network manager to manage a network. However, directory servers have not heretofore been applied, at least in the manner of the invention, to a server-based or thin client computing system such as the Citrix Systems' MetaFrame® server system, which differs fundamentally from the conventional directory service managed network not only in its ability to use thin clients, but also in that the published application servers may not be under the direct control of the network administrator, may be situated at arbitrary locations, and are accessed by an independent client associated with the thin client system rather than by the directory service or

directory services implemented program, and to which information from the directory service must be supplied. In order for a directory service to be useful in such a context, the directory service arrangement must be able to provide information on any thin client servers containing desired applications. This characteristic is known as "scalability."

The directory services enabled thin client server of the invention is not to be confused with a thin client server system that uses directory services to manage a desktop on a particular thin client server or server farm, as described in a white paper dated April 21, 1999, and entitled "*An Implementation Guide For Integrating Thin-Client Servers With The Novell's ZENworks, and NDS Products*," prepared by B. Anderson, R. Lopez, B. Calero, and E. Lee, and published on Novell, Inc.'s website at http://www.novell.com/coolsolutions/zenworks/features/a_integrating_thin_client_servers_zw.html.

The approach described in the white paper, which provides more convenient management of a particular thin client server, is not related to the present invention despite its use of a directory service in a thin client context. Instead of using directory services to publish or keep track of the thin client applications, *i.e.*, to store

authorized server lists, registration and user access information, and so forth, the prior approach uses the directory to actually manage the thin client server as a desktop, essentially replacing or forming a shell on top of the existing thin client server architecture. Thus, although it provides centralized management of a particular server, group of servers, or server farm, the prior approach completely lacks scalability, and does not address the problem of how to locate the thin client published application in the first place, or of authenticating to these published applications that may be run on different, non-centrally managed servers or server farms.

No other prior or related art is known which seeks to combine thin client servers and directory services. Background on directory services, and in particular on snap-in modules for NDS®, can be found in U.S. Patent Nos. 5,859,978, 5,987,471, and 5,991,810, while U.S. Patent Nos. 5,818,936 and 5,892,828 discuss sign-on and authentication procedures for NDS®. Other patents more generally related to directory services include U.S. Patent Nos. 5,862,325, 5,826,027, 5,913,033, 5,918,039, and 5,941,949. Server-based or thin client computing systems are described in U.S. Patent Nos. 5,826,027, 5,913,033, 5,918,039, and 5,941,949.

As thin-client technology becomes increasingly critical, the need increases for management software that can integrate thin client applications management with the more traditional applications management provided by directory service programs such as NDS® and ZENworks®. By facilitating access to the remotely run applications using directory services, the invention offers the network administrator far greater flexibility in delivering applications to users of the network, including the option of switching between locally and remotely run applications in case of a local system crash or inadequacy of hardware on the network, as well as the ability to consolidate reporting and auditing functions for both the locally and remotely run applications, and to provide end-users with common access to both locally and remotely run applications using the same relatively simple sign-on and authentication procedures, in a manner that can be made completely transparent to the end-users.

SUMMARY OF THE INVENTION

It is accordingly a first objective of the invention to provide a system and method for facilitating access to application programs of the type run on a thin client server, by using a directory service to publish the application programs, and by supplying information on the

published application programs to a client that provides access to the thin client application program using the directory.

It is a second objective of the invention to enable a thin client server system, such as the Citrix Systems MetaFrame® system, to utilize directory services to advertise their applications by having them published on the directory services instead of relying on the currently used broadcast technology, and to enable location of their applications without the need to refer to conventional broadcast services, thereby extending the availability of MetaFrame® published applications and other thin client applications over open networks such as the Internet, to which the currently used broadcast technology is not applicable.

It is a third objective of the invention to provide a system and method for facilitating access to application programs of the type run on a thin client server, in which security is improved by eliminating the need for users to save their credentials using permanent text based configuration files.

It is a fourth objective of the invention to enable selection of thin client servers or server farms for

running published applications which the user is authorized to use based on the time zone of the thin client server, in order to eliminate problems that arise when the thin client server is in a different time zone than the user.

It is a fifth objective of the invention to enable a network administrator to expedite the roll-out of new applications by giving the network administrator the additional roll-out option of enabling applications to be run from a thin client server when desktop does not meet requirements for pushing applications to the desktop, or following crash of a branch or local server.

It is a sixth objective of the invention to permit a user authenticated to a directory service to retrieve information about his or her authorized published applications, and the user's credentials.

It is a seventh objective of the invention to permit a user to use a single sign-on and authentication procedure for all thin client published applications as well as for local network applications managed by the directory services, eliminating the separate sign-on procedures and/or passwords currently required to access published applications on different thin client servers.

It is an eighth objective of the invention to enhance security by eliminating the permanent text based configuration files currently used to store sign-on and authentication information necessary to access published applications on a thin client server such as a MetaFrame™ server.

It is a ninth objective of the invention to bring the full functionality of a network management program such as Novell ZENworks® to a thin client computing system and, conversely, to offer the cost advantages and flexibility of thin client computing to users of a network management program such as ZENworks®.

These objectives and other objectives of the invention are achieved, in accordance with the principles of a first preferred embodiment, by providing a system and method which uses a directory service such as Novell Directory Services™ not only to manage applications stored locally on a network, but also to manage access to published application programs supplied by any number of thin client servers or server farms.

In particular, the objectives of the invention are achieved by publishing thin client or MetaFrame® published application programs in the directory services, eliminating

the need to retrieve information about the published applications from a broadcast service provided by the thin client server and yet, because the directory simply supplies information to the client that launches the thin client application, permitting implementation of the invention without the need to re-write the thin client application launching client (although modification of the application launching client for the purpose of retrieving information directly from the directory service rather than from a text-based configuration file stored on the user's computer is within the scope of the invention).

According to a first preferred embodiment of the invention, the directory is extended to publish thin client applications by a snap-in which creates server objects as well as application objects containing information necessary to access the thin client server and run the application, including lists of servers on which the thin client application may be run, and user credentials, *i.e.*, authentication and registration information.

When the user desires to run the application, the user selects the application from within the directory service implementation program, an example of which is ZENworks®, whereupon a first client associated with the thin client application, such as the Citrix Systems ICA client, reads

a temporary text-based configuration file, or appsrv.ini file. The temporary text-based configuration file may include the thin client server address and the published application name, and is created on the fly by a pre-launched second client arranged to retrieve the necessary information from the directory service in order to supply it to the first client.

The text-based configuration file may optionally then be deleted by the second client, and information on the start time of the application, the thin client server name, the user name, the user account number, the service level associated with the user, and the application recorded in a database while the thin client application is run in conventional fashion. In addition, the second client records the time when application usage is terminated in order to monitor usage time, or record a premature termination from an event log.

Alternatively, according to the principles of a second preferred embodiment of the invention, the first client that launches the application may be modified to retrieve the sign-on information directly from the directory services, without the need for creation of a temporary text based configuration file. This eliminates the need to use a pre-launched second client to supply the information to

the first client, although a second client could still be provided for other purposes, such as recording start and stop times and related information.

BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1 is a schematic diagram of a MetaFrame® environment to which the present application is applicable.

Fig. 2 is a schematic diagram of a directory-enabled MetaFrame® system arranged in accordance with the principles of the preferred embodiment of the invention.

Fig. 3A is a flowchart illustrating a thin client application sign-on procedure according to the principles of a first preferred embodiment of the invention.

Fig. 3B is a flowchart illustrating a thin client application sign-on procedure according to a second preferred embodiment of the invention.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

The detailed description set forth below describes a preferred embodiment of the invention in connection with Citrix Systems MetaFrame® server and MetaFrame® published

applications, and also in connection with Novell Directory Services™ and ZENworks®. Those skilled in the art will appreciate, however, that the MetaFrame® server is cited as an example of one type of server-based computing system or thin-client server, and that the invention is not to be limited to the Citrix Systems thin client server system. In addition, those skilled in the art will appreciate that NDS® is but one type of directory services, and that the invention is not to be limited to any particular directory services.

As shown in Fig. 2, the system of the invention includes a server or server farm 10 of a MetaFrame® or other server-based computing system and an NDS® server 11 running directory services. Server 11 is accessed through an administrator PC 12 running an administration program such as NWADMIN or ZENworks® with a snap-in that adds a MetaFrame® server object, *i.e.*, the software necessary to publish applications so that they can be accessed and run on the server or server farm 10, to the objects in the directory services directory tree. The creation of the MetaFrame® server object in the directory services permits the administrator to create published applications and distribute them to remote servers so that the applications can be run, and also to easily update the directory tree to reflect the newly published applications or any changes to

previously published applications. At the same time, the administration program associates published applications with end-users or groups of end-users and/or their computing devices 13.

In the meantime, information on the published applications is also supplied to a service running on each thin client server or server farm on which the published applications are to be run, and the local registry of the thin client servers is updated with the proper information so that the thin client servers will respond to sign-on requests. The actual applications to be run may be distributed by a directory services based network management program such as ZENworks®.

With this system, and the corresponding method of configuring the directory services to publish thin client applications in the directory tree, and associating the published applications to end users, the invention permits the network manager to manage thin client applications as easily as other network applications, to roll out new applications and even to seamlessly switch between a thin client application and a corresponding network application, for example upon failure of the network application, with all arbitration being carried out at the network level. The system and method of the invention also allows thin

client applications servers to be selected based on such criteria as the time zone of the server.

When an end-user signs on to directory services, the end-user is authenticated by directory services, which provides the end-user with an application launcher that enables the end-user to launch any applications to which the end-user is authorized without requiring separate authentication for each application to be launched. In the case of a thin client application, the application launcher serves to retrieve from the directory the information necessary to contact and run the thin client application, using a first client program associated with the thin client application server system.

In a first preferred embodiment of the invention, as illustrated in Fig. 3A, when an end-user wishes to access a published application, the end-user must first sign-on to the directory service (step 100), which authenticates the user and presents the user with a screen (element 20 in Fig. 2) that includes icons for each available application. Upon selecting an application (step 110), the directory service or directory service management program pre-launches a client that retrieves information on the selected published application from the directory service, including the location of thin client servers running the

application, and any necessary passwords, i.d. numbers, and other registration and authentication information (step 120) and writes the information to a temporary text-based configuration file such as appsrv.ini (step 130). The directory service or directory service management program then launches the appropriate thin client application launching client, such as the Citrix Systems ICA client (step 140), which refers to the appsrv.ini file in its usual manner in order to log on to and run the thin client application (step 150).

Following sign-on, the system and method of this embodiment of the invention deletes the appsrv.ini file, which is no longer needed since the information is still available in the directory service for the next sign-on (step 160), and records in a database the start time of the application, the thin client server name, the user name, the user account number, the service level associated with the user, and the application (step 170). Subsequently, the pre-launched second client, which is now post-launched, records the time the application usage is terminated, or retrieves the usage termination time from the event log of the snap-in server object and records it in the database (step 180).

Alternatively, as shown in Fig. 3B, the thin client application launching client may be modified such that, after performing steps 100 and 110, steps 120 and 130 are replaced by the step 190 of directly retrieving the sign-on and authentication information from the directory service, before performing steps 140-180.

In either case, the operation of the various clients is transparent to the end-user, who simply selects the desired application and utilizes it as if it were running locally on the end-user's own computing device.

Having thus described a preferred embodiment of the invention in sufficient detail to enable those skilled in the art to make and use the invention, it will nevertheless be appreciated that numerous variations and modifications of the illustrated embodiment may be made without departing from the spirit of the invention. Accordingly, it is intended that the invention not be limited by the above description or accompanying drawings, but that it be defined solely in accordance with the appended claims.

What is claimed is:

1. A server-based computing system with integrated directory services, comprising:

a module for a directory services administration program, said module being arranged to store information concerning at least one thin client published application to be run on a thin client server, including a location of the server and user registration and authentication information; and

a first client program that signs-on to the application,

wherein said first client program utilizes said stored information to accomplish sign-on to the application.

2. A system as claimed in claim 1, wherein said first client program is a Citrix Systems ICA® client, and said module includes Citrix Systems MetaFrame™ server software arranged to publish said application.

3. A system as claimed in claim 2, wherein said directory service is Novell Directory Services™.

4. A system as claimed in claim 3, wherein said module is a Novell ZENworks® module.

5. A system as claimed in claim 1, wherein said directory service is Novell Directory Services™ and said module is a Novell ZENworks® module.

6. A system as claimed in claim 1, wherein said information is retrieved from said directory service by a pre-launched second client program that writes the information to a text-based configuration file for use by said first client program.

7. A system as claimed in claim 6, wherein said second client program deletes said text-based configuration file following sign-on.

8. A system as claimed in claim 1, wherein said first client program retrieves said information directly from said directory service.

9. A system as claimed in claim 1, wherein a location of said thin client server is selected to be in a same time zone as an end-user.

10. A method of integrating a server-based computing system with directory services, comprising the steps of:

arranging a module in a directory services administration program to store information concerning at least one thin client published application to be run on a thin client server, said information including a location of the server and user registration and authentication information; and

causing a first client program to sign-on to the application by utilizing said stored information to accomplish sign-on to the application.

11. A method as claimed in claim 10, wherein said first client program is a Citrix Systems ICA® client, and said module includes Citrix Systems MetaFrame® server software arranged to publish said application.

12. A method as claimed in claim 11, wherein said directory service is Novell Directory Services™.

13. A method as claimed in claim 12, wherein said module is a Novell ZENworks® module.

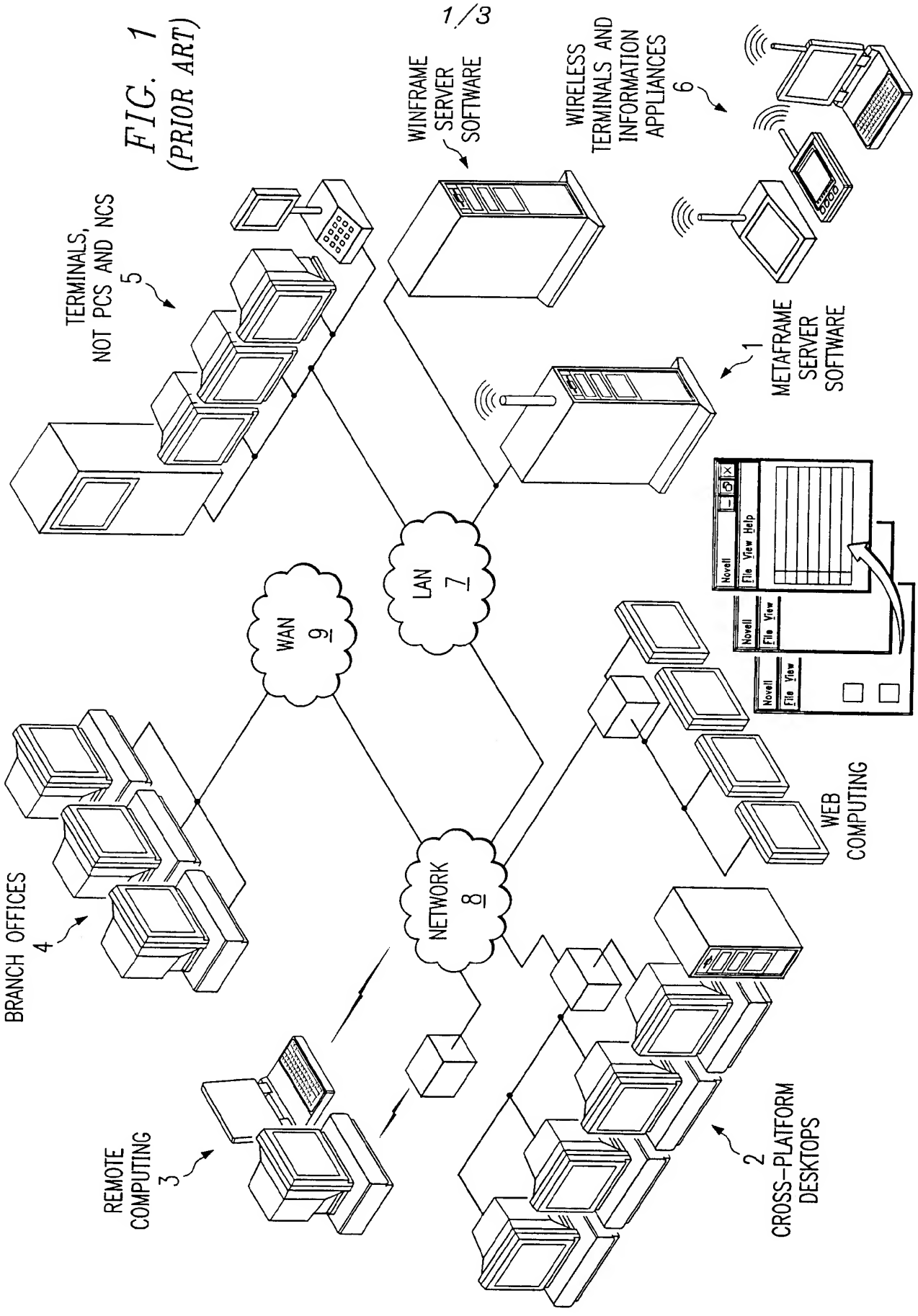
14. A method as claimed in claim 10, wherein said directory service is Novell Directory Services™ and said module is a Novell ZENworks® module.

15. A method as claimed in claim 10, further comprising the steps of pre-launching a second client program to retrieve information from said directory service and write the information to a text-based configuration file for use by said first client program.

16. A method as claimed in claim 15, further comprising the step of deleting said text-based configuration file following sign-on.

17. A method as claimed in claim 10, wherein said first client program retrieves said information directly from said directory service.

FIG. 1
(PRIOR ART)



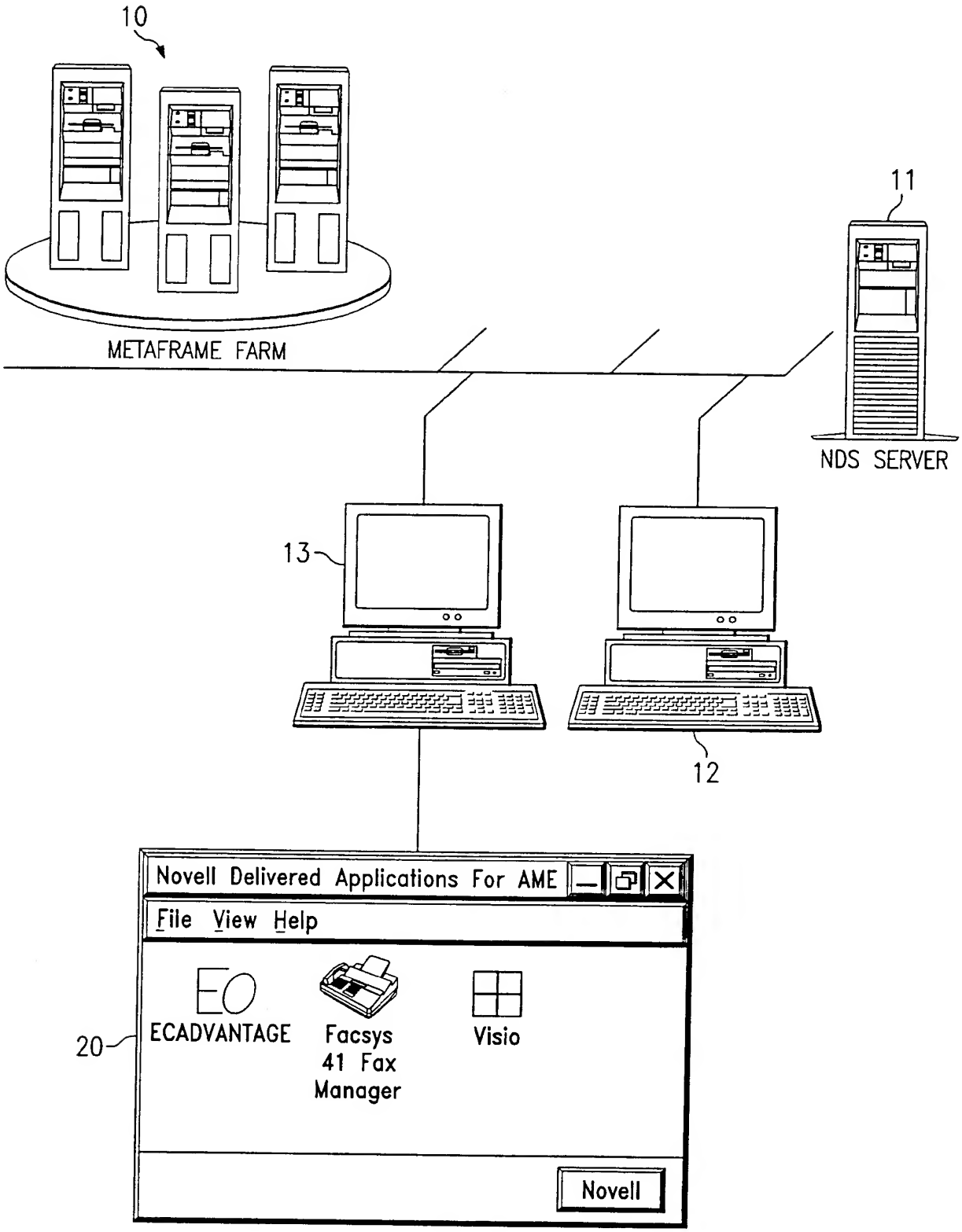

10
METAFRAME FARM
11
NDS SERVER
13
12
20
Novell Delivered Applications For AME
File View Help
ECADVANTAGE
Facsys
41 Fax
Manager
Visio
Novell

FIG. 2

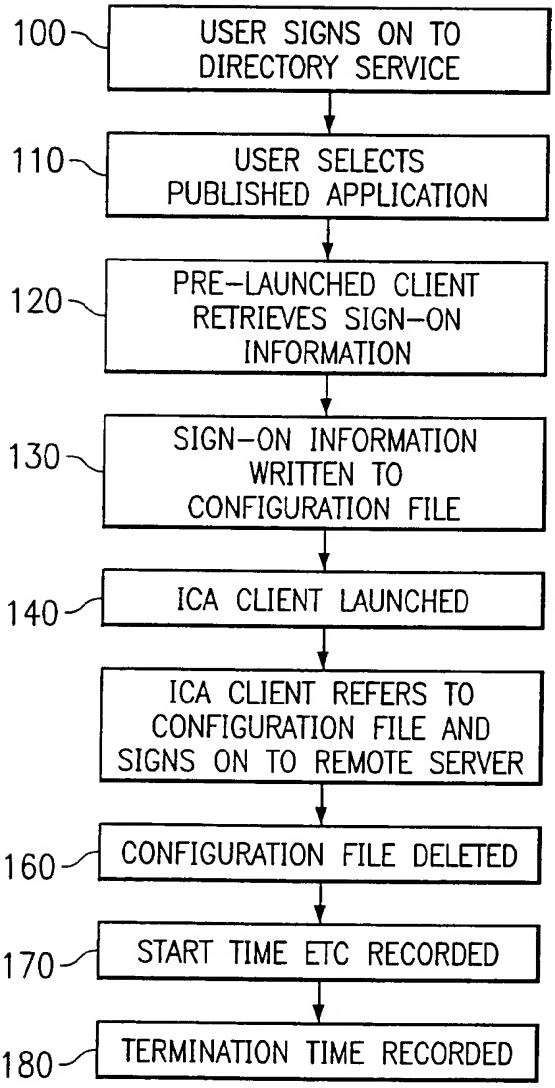

100
USER SIGNS ON TO DIRECTORY SERVICE
110
USER SELECTS PUBLISHED APPLICATION
120
PRE-LAUNCHED CLIENT RETRIEVES SIGN-ON INFORMATION
130
SIGN-ON INFORMATION WRITTEN TO CONFIGURATION FILE
140
ICA CLIENT LAUNCHED
ICA CLIENT REFERS TO CONFIGURATION FILE AND SIGNS ON TO REMOTE SERVER
160
CONFIGURATION FILE DELETED
170
START TIME ETC RECORDED
180
TERMINATION TIME RECORDED

Fig. 3A

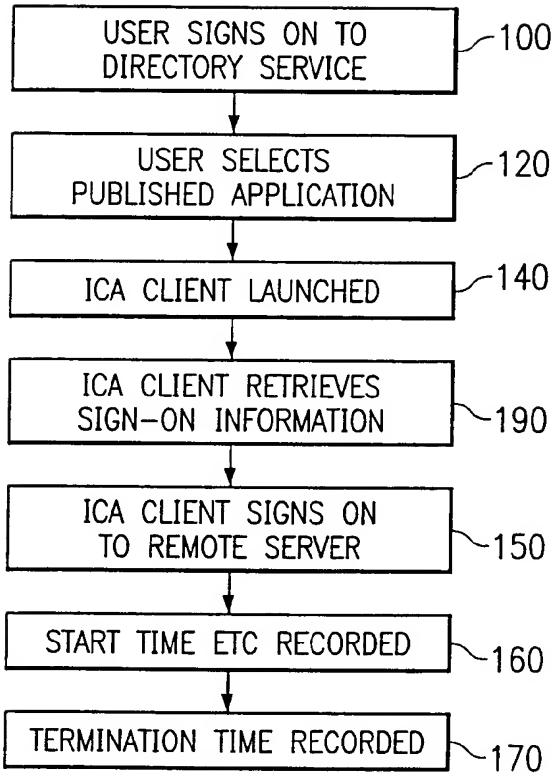

USER SIGNS ON TO DIRECTORY SERVICE
100
USER SELECTS PUBLISHED APPLICATION
120
ICA CLIENT LAUNCHED
140
ICA CLIENT RETRIEVES SIGN-ON INFORMATION
190
ICA CLIENT SIGNS ON TO REMOTE SERVER
150
START TIME ETC RECORDED
160
TERMINATION TIME RECORDED
170

Fig. 3B